パターンとペアプロの数式処理ソフト学習への適用 (数式処理と教育)

AUTHOR(S):

西谷, 滋人; 廣岡, 愛未

CITATION:

西谷, 滋人 ...[et al]. パターンとペアプロの数式処理ソフト学習への適用 (数式処理と教育). 数理解析研究所講究録 2011, 1735: 127-139

ISSUE DATE:

2011-04

URL:

http://hdl.handle.net/2433/170795

数理解析研究所講究録
第1735巻 2011年 127-139

パターンとペアプロの数式処理ソフト学習への適用

関西学院大学　理工学部（・情報科学科）　西谷滋人 (Shigeto R. Nishitani)
Department of Informatics
Kwansei Gakuin University
関西学院大学　理工学部（・学生）　廣岡愛未 (Emi Hirooka)
現・高砂市立荒井中学校（・非常勤職員）
Arai Junior High School

1　まえがき

数学のデモンストレーションをしていて学生たちの顔を見ると，数年前にみた息子の友達の笑顔を思い出します．皿に野菜を盛ったときちょっと困った様子で,「あなたのおっしゃりたいことはわかるし，私も何度も努力したのですが，だめです．食べられません」とつぶらな瞳で訴えていました．

大学で数学教師が学生に教えるときは，そういう経験をいやほどしてきたのかもしれない学生を説得する必要があります．彼らにとって演習は修行ではなく，単なる苦役でしかないようです．もっとも精神を萎えさせる刑務所での苦役である，穴の埋め戻しと，答えがわかっている問題を解くのの違いをいくら訴えても学生には伝わりません．そんな学生は演習の値切りにどっぷり浸かってます．習慣を変えるスイッチを入れるためには，相当巧妙な説得が必要です．ハック本や映画にあるような「AHA!」体験は滅多に起こらないし，英会話のように成果が手軽に得れるわけではないので，演習の継続を強制するようなシステムが必要なのですが，そんなにうまいシステムは今のところなさそう．したがって，地道に旧来の説得を続けるしかありません.「ほらこんなに役に立ちますよ.」とか「ほらこんなに簡単で，便利ですよ.」てね.

さて，そんな中で，数学とは少し畑の違ったプログラミングの世界，ソフトウェア工学なる領域で，役に立つとされる二つの手法について紹介し，数式処理ソフトMapleの学習に適用した結果について報告します．二つの手法とは,「デザイン・パターン」と「ペアプログラミング」．そしてまだ4年生の研究室レベルでの取り組みなのですが,「フロー状態」について報告します．

対象とする授業は情報科学科3年生に提供している数式処理演習．週一コマ・半期で，コンピュータが一人一台用意された部屋で提供しています．必修ではないが選択必修の4科目の1つ．だけど，20いくつかの演習から選択できるうちの1つでしかありません．でも，8割の学生が「数学を苦手とする学生を対象とする」と言う宣伝に惹かれて履修してます．従って学生・教官双方に数式処理ソフトのスキルに対して，どうしても定着させなければならないスキルという悲壮感がないのが問題です．数式処理ソフトだけでは

ないのですが，何かのスキルを定着させるのに一番いい手法は，必須とすることでしょう．全員が履修して，授業で複数の先生が標準として使う．そうすれば，必然的に覚えざるを得ません．これを関西学院大の数理科学科ではやろうとしています．どうなるか愉しみです．

2 パターン

さてまずパターンについて．ソフトウェア工学というのはなじみがないかもしれませんが，大きなコードを間違いなく開発，運用する手法．成果物としては，オブジェクト指向とかウィキペディアとか，XP(Extreme Programming) なんかを知っているかもしれません．これらの源流をたどるとケンブリッジ大の数学科に進んで後に建築家になったクリストファー・アレグザンダーに行き着きます [1]．1977 年に著した「パタン・ランゲージ」では，都市・町並み・家・部屋などの建築の全ての構成物を形式・粒度によって分類し，「無名の質」という美意識を基準にカタログとして提供しています [2]．

アレグザンダーが提案した知識構築への考え方は，複雑な全体像を理解，記憶，利用しやすいように，「階層構造，細かな分類，相互参照」など構造的にすることで，小さい部分から見始めても，大きな全体像から見ても理解できるようになっています．さらに，図 1 に示したように体裁を整えて「名称，写真や図，上位参照，問題，考察，解答，下位参照」が定まった順序に提供されています．これだけではわかりにくいですが，図 2 にまとめ直したように，粒度の上下項目との関連や，問題，一目で分かるイラスト，箇条書きのまとめ等によって覚えやすい工夫がなされています．粒度や上下参照はまさに頭の中にある知識構造を表出させたようなイメージを与えてくれます．木構造のような，単一のルートによって上下関係や階層構造が定められているのでないというのが肝です．あるいは，「逆引き」という形態での知識提供の源流なのかもしれません．

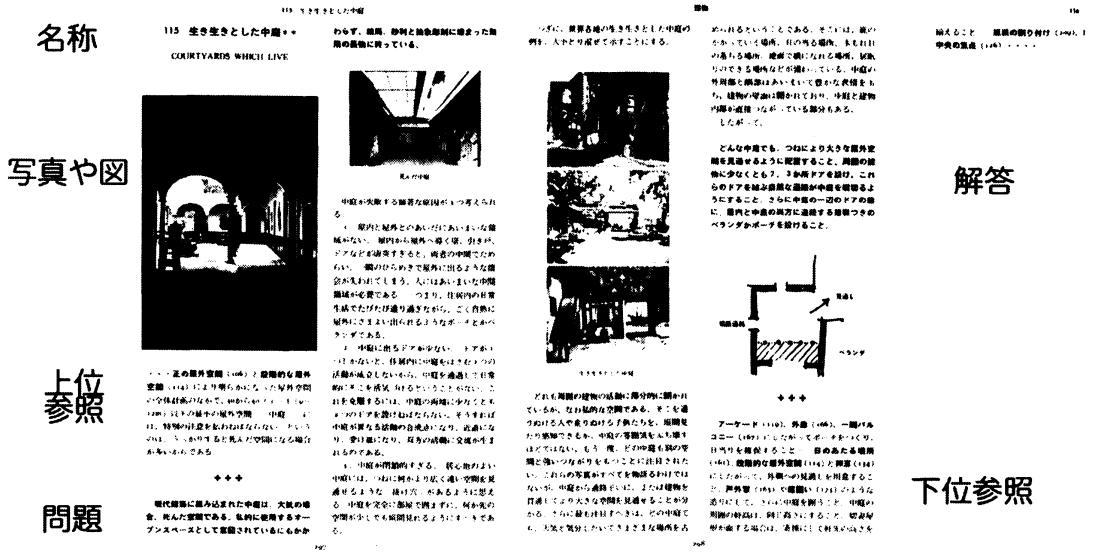

図 1: パタン・ランゲージの「No.115:生き生きとした中庭」を例にした体裁 [2]．

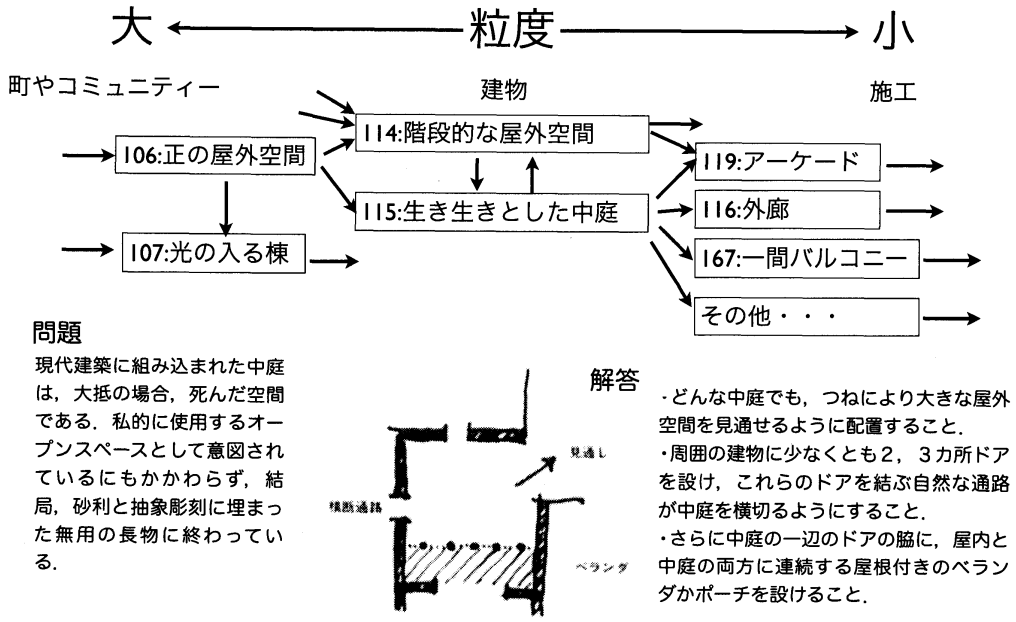

図 2: パタン・ランゲージの「No.115:生き生きとした中庭」を中心にして，粒度，上下参照の模式図，問題，覚えやすいイラスト，箇条書きの解答．

この建築業界の設計・施工の集大成をメタファとして，複雑化を極めた大規模ソフトの開発におけるチームでの設計・コード化の作業を分解・分類し，工学として昇華しようとしました．それが『デザイン・パターン』です [3]．オブジェクト指向言語 Ruby の開発者として有名な松本は，その著書で

> 「『デザイン・パターン』を扱った書籍が出た時の最初の印象は「おおげさに話題になっているわりには当たり前の内容だなぁ」というものでした．＜中略＞　しかし，しばらくしてから気がつきました．『デザイン・パターン』の本質は今までに使ったこともない新しいパターンを紹介するのではなく，しばしば使われる実用的なパターンに適切な名前を提供することによって，デザインにおける語彙を提供することにあるのだと．」

と記しています [4]．さらに，ソフトウェアに繰り返し登場するパターンをカタログとして提示し，名称を与えることで，今までは経験を積んだプログラマにしか取り扱えなかった暗黙知を，普通のプログラマにも扱えるよう形式知化した功績を賞賛しています．

『デザイン・パターン』の最初に図３に示したようなリンク構造や，カタログとしての記述項目並びにその使用法が示されており，まさに知識提供の手法として『パタン・ランゲージ』を踏襲しています．

さらに，『デザイン・パターン』提唱の首謀者のひとりであったウォード・カニンガ

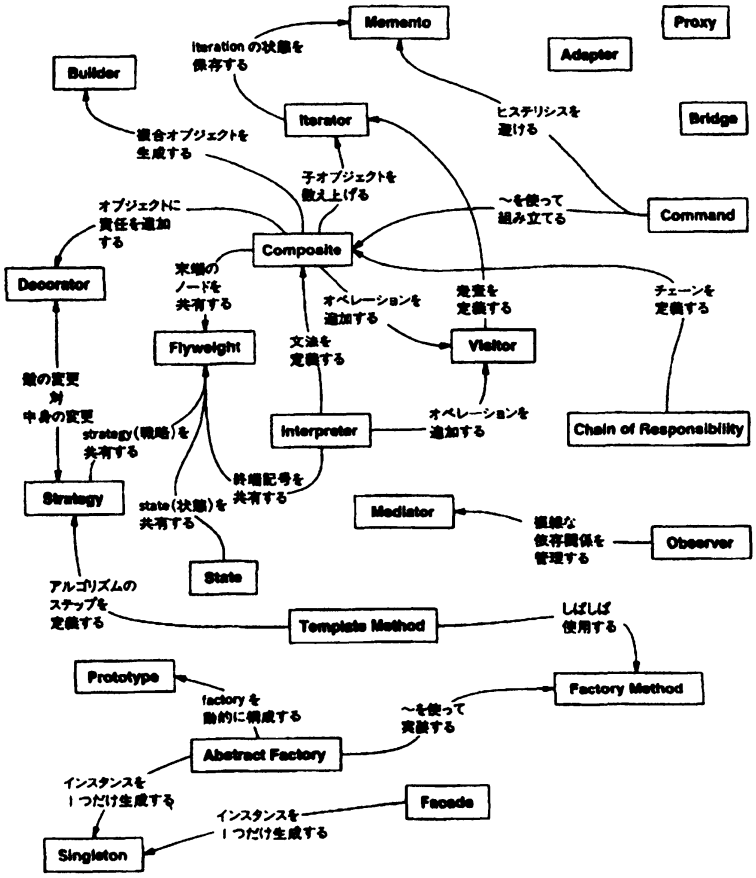

図 3: 23 のデザイン・パターンの相互関係 [3].

ムのこだわりは，構造化という中身だけではなく，カタログの提示の仕方にもありました．彼は，書籍という古い媒体ではなく，より使いやすい形態での提供を求めていたようで，一時期は同僚のビル・アトキンソンが作った HyperCard での提供を試みていました．HyperCard の net 版となる WEB が開発されるにいたり，その威力を利用するソフトとして，WikiwikiWeb をあみ出しました．これはその後，Wiki と呼ばれるソフトのジャンルを生み出し，提供する内容を「デザイン・パターン」だけにとらわれず,「知識全般」を扱うように改良されました．これをジミー・ウェールズらが広い知識を共有するプラットホームとして整備した Wikipedia は，知識検索でのデフォルトサイトとなっています [1].

このような歴史をもったパターンなんですが，我々『理系・日本人』はもっと昔からこの形態になじみがあります．初版が昭和初年，チャート式代数学とされている，数研出版の「チャート式」です [5].「見出し，例題，解法，チャート，練習」というのはまさにパタン・ランゲージの体裁を構成する要素そのものです.「チャート式」は 1 ページにこれらの要素が見事に集約されています．一方パタン・ランゲージの体裁は図 1 に示した通り，それほどまとまっていません．訳のせいかと原本に当たったのですが，もっと統一感がなく，長さがばらばらでした．これは粒度を対象から決めて，もっとも理解し

やすいブロックにまとめたからでしょう．一方,「チャート式」の粒度は，知識を受け入れる受験生にとことん合わせています．絵巻物から浮世絵の伝統が生かされています．私が大学に入った当時は，これほどきれいにまとまった演習書がなく，やる気をなくした覚えがあります．一方，現在は大学向けにこの手の演習書を，サイエンス社が精力的に出しています．特に数学では寺田文行の一連の演習書が出色です [6]．高校の演習書はさらに進んでいます．見開き２ページや色刷り，答えの配置が工夫されています [7]．いずれにしろ，知識はあらかじめ整理して提供するのが一番効率よく頭に入り，さらに演習には計画をたてて継続していくために適度な大きさにそろえるということは最低限必要な体裁と考えられます．

このような「チャート式」思想に基づいて作成したテキストの例を図４に示します．ようやく体裁の形が固まってきたところです．「チャート式」の演習テキストの作成を2006年に提案して，いくつかのversionを作ってきたんですが [8]，パタン・ランゲージの分析を通じて，粒度，相互参照についてよりパターン化した階層構造が重要であることがわかり，現在改訂中です．今までに，出版を目指して何度か出版社と相談したのですが，多色刷りやページ数などの制約が重しとなってます．多色刷りはもっとも重要な要件の１つで，プログラミングエディターでは必須です．単語レベルのカラー化によって，長い単調なコードを読みやすくしたり，文字の固まりを絵として右脳で捉える効果が指摘され，集中力を継続するために不可欠な要素となっています．また，texやwikiでの作成を試みましたが，Maple scriptの実行検証が煩雑なんであきらめました．現在はMapleに標準装備のワークシートモードを使ってpdf保存して，WEBでのダウンロードや，カラープリンターで出力し学生へ配布しています．

でもこれだけでは，定着するところまで行き着きません．この原因は演習不足なんですが，単位と言う最終のムチをできるだけ使わずに，演習をどうやって強制させようかというのが次のお話．

3　ペアプロ

成績はそれほど良くないのだけど，なかなかするどい女子学生が私のところに来て,「寡黙に自分との戦いみたいにプログラムを組むのが嫌だ」と言うのです．実際にプログラミングを学習した学生が言うのだから，やっぱりプログラマーに対して一般的に持たれているわかりやすいイメージがこれなんでしょう．数学の演習課題をやるのも同じようなイメージがあります．実際にプログラムを教える時や，課題を解く時，自分で新しいプログラミング言語や手法を習得しようとする時はまったく違った作業をしているのだけど... ペアプロがその寡黙でない楽しい１つの形です．

プログラマーの生態をとてもよく表わす記述が，ソフトウェア開発者に人気のブログを書籍化した「Joel on software」にあります [9]．

> ひとたびフロー状態になると，それを維持するのは難しくない．私の一日の多くはこんな感じだ: (1) 仕事にとりかかる．(2) emailをチェックしたり，Webを見たり，その他のことをする．(3) 仕事に取りかかる前にランチを取った方がいいと判断する．(4) ランチから戻る．(5) emailをチェックし

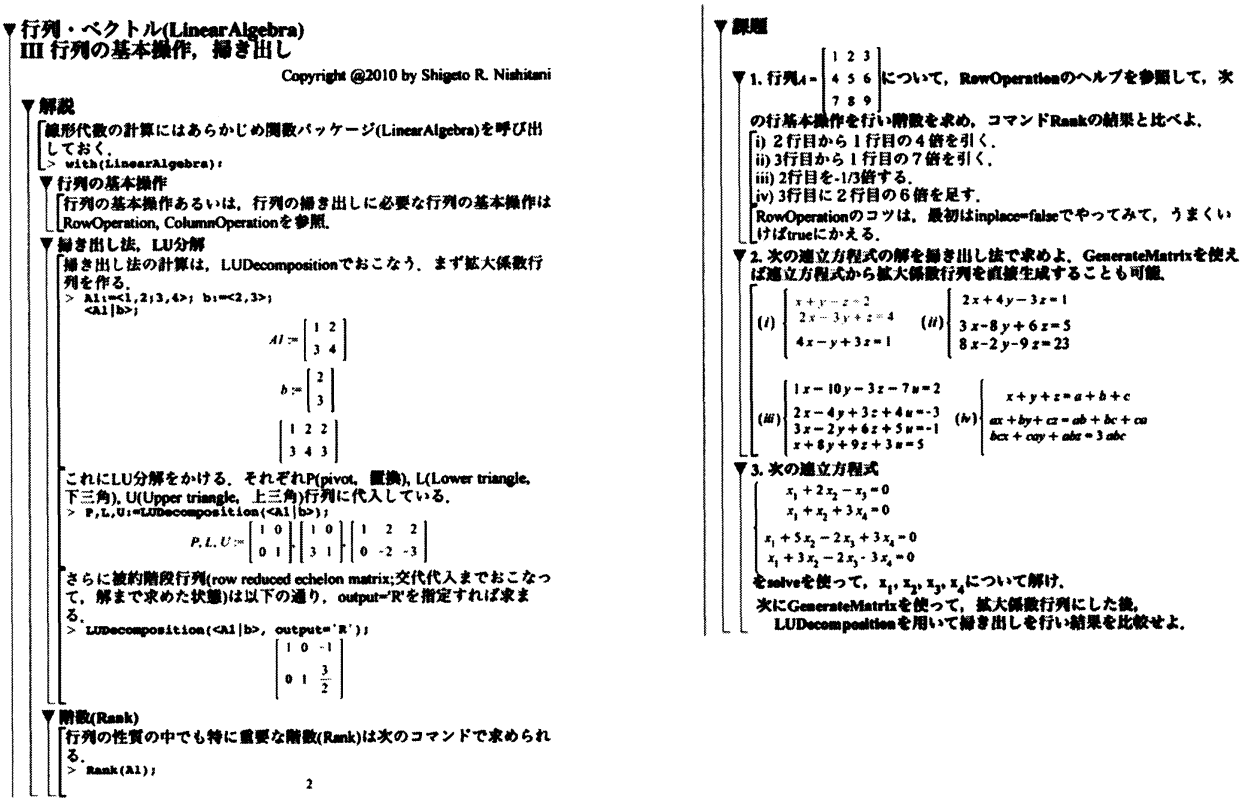

図 4: チャート式の体裁にならったMaple入門用テキストの見本.

> たり，Webを見たり，その他のことをする．(6) いい加減始めた方がいいと心を決める．(7) emailをチェックしたり，Webを見たり，そのほかのことをする．(8) **本当に**始めなきゃいけないと，再び決心する．(9) くそエディタを立ち上げる．(10) ノンストップでコードを書いていると，いつのまにか午後7:30になっている．

職業人としてのプロのプログラマーでさえこうなのですから，プログラミングの修行を始めたばかりの学生さんたちが集中して，演習に取り組めないのは仕方がないことなのかもしれません．ネットのある環境では，どうしても自分で解くよりも答えを探したがります．数式処理ソフトのMapleを教えるのも同じ傾向があります．Mapleは幸いにして，あまり使える人がいないので，グーグル先生に聞いても答えてくれません．あるいは英語だったりします．大学で身に付けた最先端の問題解決法であるグーグル先生から離れられずに，試験中でも役に立たないサイトを必死に見ています．テキストに答えがあるのに...

このような状況を打破したくて導入したのが，ペアプロです．先ほどのJoelの言葉は

> **ただ始めること**，たぶんこれが生産性の鍵なのだ．ペアプロが機能する理由は，相方とペアプロ作業の予定を立てるときに，お互いに始めることを強要するからに違いない（原文より訳出）．

と続いています．彼もペアプロの効力を認めています．ではペアプロとはどんな技法な

のでしょうか.

ケント・ベックらが開発したEXやAgileと呼ばれるソフトウエア開発法のなかで推奨されている方法で，プログラミング効率を劇的にあげるとされています[10]. やり方は簡単で「ふたりで，共同作業しましょう」という結婚式のノリです. 一台のコンピュータで，モニタ，キーボード，マウスも交互に使って，プログラムを書いていきます. 5分ほどで交代. 疲れたら交代. 分からんようになったら交代.「ドライバーとナビゲータ」,「ぼけとつっこみ」の要領で作業を進めていきます.

さて，このようなペアプロのメリットは以下の通りです.

単純なミスのチェック：変数の意味，括弧の閉じ忘れ，セミコロン忘れのチェック.

気づき：先生が提示してしまうと単なる暗記になる手順を，共有することによって発見の疑似体験ができるのではないかという期待. コマンド一発で済む計算を組み合わせて，複数ステップの数学問題の解答を構築する過程の共有です.

コミュニケーション：理系と文系の違いなんですが,「どう思う」と聞かれたときには「答えをさがす」くせがついてしまっていて，沈思黙考してしまいます. これが面接では大失点になります. 思考過程を言語化して他人に説明する訓練となります.

集中（キックオフ）：一人でいるとついふらふらとネットサーフィンしたりメールを書いたりしますが，共同で考えているときにはこのようなことができず，課題に集中することが期待できます. Joelが指摘しているペアプロの最大の効用がこれです.

かっこづけ：人が見ていると，わかりやすい形に仕上げるちょっとした手間をかけようかという気持ちが発生します. これが後で（自分で）見ても理解しやすいコードを書くコツです.

このような作業を通じて，一人では挫折しがちなスキル習得を二人の共同作業として完遂してもらおうという意図です. たしかに演習中にはよく会話がなされ，わいわい言いながらやっている光景がそこかしこに見られます. また，集中してレポートを仕上げる努力もなされているようです. 章末に示したような最終試験を課していますが，図5に示したように最終的には2/3の学生が80点を取っています. しかしこれがペアプロの成果であるかは，比較実験をしたわけではないので不明です. ただ，個人での作業を強いていた時よりも学生たちが嬉々としてやっているのだけは確かです. また，研究室に進んだときに自然にペアを作ってプログラミングを進めてくれるようになりました. そっち方面の技能の習得は素早いです.

ペアの組み方についてはランダムペア(07年度)，半期ランダムペア(08年度)，半期4人組(09年度)，4人組(10年度)などと，いろいろ試みましたが最適解は見つかっていません. ペアの実力が均衡している場合は議論や教え合い・学び合いが生まれるのですが，差があるとうまくいきません. できる子ができない子に教えつくしていないとか，できる子に全て任せてできない子は頭を使わない「フリーライダー（ただ乗り者）」になっ

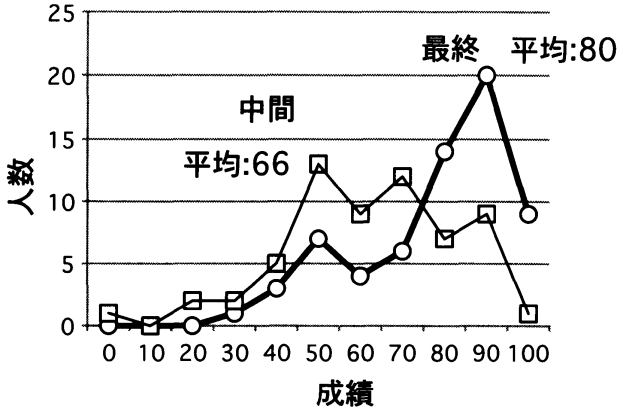

図 5: 試験の得点分布.

ています．できる子は数式処理のコマンドを覚えてなくてもできてしまい使用法を忘れてしまいます．一方，できない子は日常的な演習と言う習慣の定着には至らず，使用法がわからずに適当にいじくり回して時間・期間終了という感じがあります．数式処理ソフトのレポートは，自力で解いたかどうかが信用できないので，必然的に試験による個人の評価となります．グループ課題での取り組みをどう評価するかは未解決です [11].

4　まとめ，そしてフロー状態へ

ところで最近の理系大学生はチャート式を知らないのをご存知？　やらなかったではなく，知らないの．したがって,「チャート式」風と言ってもそれだけではピンと来ません．教員の間でも「チャート式」と言ったときに，単に沢山演習をさせることとしか捉えてないことがありますので，あまり単語だけで「チャート式」という概念伝達を済まさない方がいいようです．いずれにしろ今時の，つまり「ゆとられた」学生は知識が手から身に付くとか，スキルとなるということの実感がないので，演習書とのつきあい方を知りません．どういう目的でその作業をするかをより具体的に指示してやらないとモニターをにらんで凍っています．スキルは断片だけでも十分機能するからと言ってそそのかしても，どうしても最初のページから知識習得風にやりたがります．これはロールプレイングゲームの弊害かも．といって最初からちゃんと読んで，覚えているかというとこれもだめ．そして，記述・構造の不十分さ，誤植などがちょっとしたつまずきとなり，途中で挫折してあきらめてしまうことが多々あります．テキストは完璧でなければならないようです．まいったな．ええかげんでええのに.

多くの学生は「やればできる」と思っています．学習の動機の基本であるこのような「しなやかな心持ち (growth mindset)」に対して,「こちこちな心持ち (fixed mindset)」の持ち主は「能力は生まれつき」と思っているので能力を向上するための学習が進みません．それぞれの心持ちによる考え方の典型例をキャロル・ドゥエックは,

心持ち　こちこち (fixed)vs しなやか (growth)

賞賛 能力を評価 vs 努力を評価

ゴール 結果ゴール（能力のお墨付きが欲しい，成功しても失敗しても不安）vs 過程ゴール（どれだけ伸びたか，否定的な感情が減っていく）

努力 能力のなさの証明 vs 成功のための必須項目

反応 (失敗した時の) 感情的になり目を背ける vs どう感じたかよりも何かを学ぼうとする

戦略 1つのやり方に執着する vs いろいろやる（とんでもないぐらい）

と分類しています [12]．でもね，その気になってやってみたけど，やっぱりだめだったという経験の果てに教室や演習室にたどり着いているのかもしれません．著名な数学者溝畑茂が「知識もテクニックもないし，自信がないんでどうしましょうか？」という問いに，「アホ，自分の知識やテクニックに頼るやつは，自信のない証拠や．」と答えたと森毅が伝えてます [13]．「なにもない自分に自信を持て」ってすごいけど，酷ですよね．持ち物に頼るのは自信がないんで，持ち物はこれからつくりゃいいのだけど，それを実行するためのスキルや心持ちが身に付いてない学生が多いのが現状です．少しやってだめならあきらめて，さらに自信をなくしちまっているだけかもしれません．動機はあると思うのですが，昔ほどではなく，スキルもおぼつきません．そのような学生に対して，再度同じチャレンジをしてみと言ってもきついかも．

近頃の学生さんはもしかして，知識とスキルの違いがわかってないのかもと疑います．知識は今やグーグル先生のおかげで，数秒のタイプで答えが出てきます．ところが，頭でなく体で覚えるスキルを身につけるには演習が不可欠です．学校や塾で「演習」をしてきたはずなのに，それはテレビを見ているのと同じで，解答が提供される様子を眺めることと同義らしい．それを，先生が示すのか，優秀な友達が示すのか，あるいは演習書の解答を見るのかはちょっとした違い．時間を決めて，あるいは量を決めて自分で解いてみる，つまり自習という経験がほぼないのかも．だって「勉強＝塾」の考え方の下では，塾が提供するすばらしいカリキュラムに参加しているだけで，自動的にツリー構造の知識習得が進行していくのだから．ゆとりのモットーを「覚えるな分かれ」と捉えるなら，ある意味で教育は成功している．覚えずにわかろうと努力した結果としての学習態度が，値切りなのだから．ではどうやって成功への道となるはずの「努力を継続」する「演習の習慣」を生み出せばいいのだろうか．

一番簡単なのは，たぶんドゥエックの分類の本質と一致すると思うのですが，目標を変えることです．さきほどの「しなやかな心持ち」は生まれつき持っている考え方と言うよりも，学習を達成するために身につけるべき心持ち，つまりスキルです．その第一歩は，学習の結果に注目するのでなく，学習の過程そのものを評価することです．Joel の言葉にあった，「ひとたびフロー状態になると，それを維持するのは難しくない．」と言っているフロー状態をご存知でしょうか．M. チクセントミハイが提唱した心理状態で，何かの作業をするときに，

1. 明確な目標

2. 迅速なフィードバック

3. スキルとチャンレジのぎりぎりの限界に挑戦する

ことによって,「集中が焦点を結び，散漫さは消滅し，時の経過と自我の感覚を失う」という状態です [14]. フロー状態を模式的に示したのが図 6 です. 単純で低いレベルのスキルでも，時間を制限したり，リズムにあわせるというチャレンジを与えることで，フロー状態に入ることができます. 続けているとスキルがあがり，図の右方向に移動し，作業が退屈になってきます. このときより難しい課題に挑戦することによって新たなフロー状態に入ります. 一方，テニスなんかの対戦型スポーツでは相手が手強いと図の上方向に移動し，不安な状態となります. しかし，それを克服するためにスキルをあげるとより高度なフロー状態に入ります. これは，音楽，読書，スポーツ，運転，座禅などで経験することができます. 私は洗濯もんたたみでも経験しています. またプログラミングは，こういう状態を経験する典型的な作業です. そして，数学の問題を解いている状態もそうでしょう.

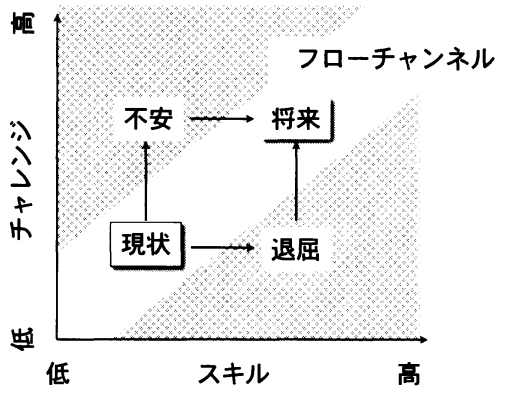

図 6: フロー状態のスキルとチャレンジの関係を示す模式図.

チクセントミハイは

> 挑戦目標の達成に取り組んでいる時が，生活の中で最も楽しい時である. 心理的エネルギーの統制を達成し，それを意識的に選びとった目標に向けた人は必然的により複雑な存在へと成長する. このような人は能力を高め，より高度な挑戦対象へと近づくことによって，しだいに非凡な能力をもつ人間に変わっていく. (p.8)

と説いています. まさに**達人への歩み**ですね.

フロー感を得るためには，チャレンジとスキルのきわどいせめぎ合い，作業に即座に反応する結果が必要です. 数式処理は計算を省いて，積分の結果やグラフをすぐに書いてくれます. これだけでも没入する要素は十分だと思うのですが... 演習をしていない学生に,「没入感」の経験がないのかと言うとそんなことはなく，ゲームがあります. 残念

ながら，私はゲームの「没入感」が全くないんです．学生時代にゲーセンでやりすぎて，金の無駄という刷り込みができてしまったようです．息子には悪いんですが，一緒には楽しめません．あっち向いてほいとか，アニメでは没入できるんだけど．フロー状態は自分なりの価値のある結果へのチャレンジでないと得られないようです．今の若い世代はゲームにも価値を見つけています．そんな学生に，数学での適切な到達目標を明確にすること，あるいは価値のある挑戦目標の設定ができれば，演習を定着させることも可能でしょう．フロー感は習得しているのだから，後はその状態へ引きずり込めばいいだけです．そのうち自分から進んで入るようになります．マラソン（脳）と同じように．

ペアプロの様子は，昔の予算がない学生実験で少ないコンピュータを取り合いして，ああでもないこうでもないと試行錯誤をしていた，あるいは，先輩がまるで魔法をかけるかのようにコードを構築していくのを横で見ながら，邪魔せん程度に「これなんすっか」て聞いていた雰囲気と同じです．その時は，コンピュータやソフトに対する畏怖や憧れが強い学習動機になって，意図せずにソフトウェア工学的には現在の最先端の学習形態を実践していたようです．しかし，動機の低い学習者には，演習を継続させる仕組みが必要で，ペアプロは1つの可能性です．ペアをどうするか，どう評価するかが解決できれば強力な道具となりそうです．

私が数学やプログラミングを学んだ時代は，今ほどカリキュラムも環境も進んでなかったけど，とてもいい時代だったなあと実感します．一方で今，目の前にある現実を解決するのに日々苦闘していると，いい解があったら教えてほしいと思っちゃう時があります．でも，「パタン・ランゲージ」は設計だけでなく，施工・改築も利用者がするように説いています．

参考文献

[1] 江渡浩一郎著,「パターン，Wiki, XP」，(技術評論社，2009).

[2] クリストファー・アレグザンダー著，平田 翰那訳「パタン・ランゲージ—環境設計の手引」，(鹿島出版会，1977).

[3] エリック ガンマ, ラルフ ジョンソン, リチャード ヘルム, ジョン ブリシディース著, 本位田 真一, 吉田 和樹訳「オブジェクト指向における再利用のためのデザインパターン」(ソフトバンククリエイティブ，1999).

[4] まつもとゆきひろ著，日経Linux編,「まつもとゆきひろ コードの世界 スーパー・プログラマになる14の思考法」(日経BP出版センター，2009) p.126.

[5] 例えば，砂田利一，柳川高明著,「チャート式　数学I+A」(数研出版，1963).

[6] 寺田文行，木村宣昭著,「演習と応用線形代数」(サイエンス社，2000).

[7] 廣岡愛未，関西学院大学理工学部卒業論文 (2010).

[8] 西谷滋人,「数式処理ソフトMAPLEによる数学教育」,「現代数理入門」宮西正宜,茨木俊秀編著,(関西学院大学出版会,2009),pp.211–231.

[9] Joel Spolsky 著,青木靖訳「Joel on software」(オーム社,2005) p.132.

[10] ローリー ウィリアムズ, ロバート ケスラー, 長瀬 嘉秀, 今野 睦監訳, テクノロジックアート訳,「ペアプログラミング—エンジニアとしての指南書」,(ピアソンエデュケーション,2003).

[11] エリザベス・バークレイ,パトリシア・クロス,クレア・メジャー著,安永悟監訳「協同学習の技法,大学教育の手引き」(ナカニシヤ出版,2009).

[12] キャロル S. ドゥエック著,今西 康子訳『「やればできる!」の研究—能力を開花させるマインドセットの力』草思社 (2008).

[13] 森毅著「東大が倒産する日」旺文社,1999,p.67.

[14] M. チクセントミハイ著,今村 浩明訳,「フロー体験 喜びの現象学」世界思想社 (1996/08)

v.10.1 2010/6/23 実施

情報科学科　数式処理演習試験問題

以下の問題を Maple で自力で解き，出力して提出せよ．80 点以上が合格．何番をやっているかが分かるようにせよ．

1. (a) $\cos 2x - \cos^2 x$ を微分せよ．(10 点)

   (b) $\sqrt{\dfrac{x^2-1}{x^2+1}+1}$ を `x=-10..10` で `plot` せよ．(10 点)

2. (a) $\displaystyle\int \frac{1}{\mathrm{e}^{2x}-2\,\mathrm{e}^{x}}dx$ を求めよ．(10 点)

   (b) $\displaystyle\int_1^2 x^n \ln(x)\,dx$ を求めよ．(10 点)

3. (a) 行列 $A=\begin{bmatrix} 1 & -1 & -1 \\ -1 & 2 & 2 \\ 2 & 1 & 2 \end{bmatrix}$ の逆行列を求めよ．(10 点)

   (b) 行列 $B=\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & 2 \\ 0 & 2 & 1 \end{bmatrix}$ の固有値と固有ベクトルを求めよ．また，各固有ベクトルがお互いに直交していることを確かめよ．(10 点)

4. a, b を定数とし，$a \neq 0$ とする．2 次関数

$$y = ax^2 - bx - a + b$$

   のグラフが点 $(-2, 6)$ を通るとする．

   このとき

$$b = -a + \boxed{\text{ア}}$$

   であり，グラフの頂点の座標を a を用いて表すと

$$\left(\frac{-a+\boxed{\text{イ}}}{\boxed{\text{ウ}}\,a},\ \frac{-\left(\boxed{\text{エ}}\,a-\boxed{\text{オ}}\right)^2}{\boxed{\text{カ}}\,a} \right)$$

   である (2008 年度大学入試センター試験数学 I より抜粋)．これらの結果を Maple で求めよ．(20 点)

5. ドリーン・アンドリカは，n 番目の素数を p_n とすると，すべての n に対して $\sqrt{p_{n+1}}-\sqrt{p_n} < 1$ が成立すると予想している．n を 1000 未満としたとき，最も大きな差は $\sqrt{11}-\sqrt{7} = 0.670873\cdots$ であり，確かに 1 を下回っている．このことを確かめるプログラムを作れ．(20 点)

   ヒント：i 番目の素数は `ithprime(i)` で求められる．